

# Enhancing Network Visibility and Security through Tensor Analysis

Muthu M Baskaran, Tom Henretty, James Ezick, Richard Lethin

*Reservoir Labs Inc., 632 Broadway Suite 803, New York, NY 10012*

David Bruns-Smith

*University of California, Berkeley, CA*

---

## Abstract

The increasing size, variety, rate of growth and change, and complexity of network data has warranted advanced network analysis and services. Tools that provide automated analysis through traditional or advanced signature-based systems or machine learning classifiers suffer from practical difficulties. These tools fail to provide comprehensive and contextual insights into the network when put to practical use in operational cyber security. In this paper, we present an effective tool for network security and traffic analysis that uses high-performance data analytics based on a class of unsupervised learning algorithms called tensor decompositions. The tool aims to provide a scalable analysis of the network traffic data and also reduce the cognitive load of network analysts and be network-expert-friendly by presenting clear and actionable insights into the network.

In this paper, we demonstrate the successful use of the tool in two completely diverse operational cyber security environments, namely, (1) security operations center (SOC) for the SCinet network at SC16 - The International Conference for High Performance Computing, Networking, Storage and Analysis and (2) Reservoir Labs' Local Area Network (LAN). In each of these environments, we produce actionable results for cyber security specialists including (but not limited to) (1) finding malicious network traffic involving internal and external attackers using port scans, SSH brute forcing, and NTP amplification attacks, (2) uncovering obfuscated network threats such as data exfiltration using DNS port and using ICMP traffic, and (3) finding network misconfiguration and performance degradation patterns.

### *Keywords:*

Network analysis, Cyber security, Tensor decompositions, Network threats

---

## 1. Introduction

Network analysis and network threat identification are notoriously difficult problems to solve. Traditional signature-based approaches are often thwarted by the ever-changing nature of modern cyber threats. It is nearly impossible to define signatures for what is or is not normal that generalize across many networks. Even on a given network, expected behaviors might change from day to day.

Furthermore, it might not be possible to write coherent rules that capture all activities of concern.

The application of cutting-edge data analytics to network traffic logs has struggled to surpass the shortcomings of classical signature-based systems. Supervised techniques run afoul of the same key problem – it is not realistic to specify normal versus abnormal behavior upfront. Other approaches that rely on training a model based on large volumes of historical data are hindered by another issue – because of the sensitive nature of network traffic there is very little publicly-available training data, and that data is not guaranteed to generalize in a meaningful way to the user's own network.

Tensor decompositions are a class of algorithms

---

### *Email addresses:*

{baskaran,henretty,ezick,lethin}@reservoir.com  
(Muthu M Baskaran, Tom Henretty, James Ezick, Richard Lethin), bruns-smith@berkeley.edu (David Bruns-Smith)

that provides a new approach for analyzing network traffic data that has been demonstrated to overcome these traditional shortcomings. A tensor is a multidimensional array of data – a suitable abstraction for structured network metadata collected in the form of network logs. A tensor decomposition breaks down a tensor, such as a log, into a finite set of patterns, called components. In this way, tensor decompositions perform a form of unsupervised learning on network traffic that does not require prior training data.

A tensor-based approach is uniquely better suited than other classical unsupervised machine learning approaches for network analysis and cyber security in that a tensor decomposition can natively capture patterns that span the entire multidimensional data space. This can include patterns that reflect multiple sources, multiple receivers, periodic time intervals, and other complex patterns that cannot be captured with approaches such as k-means clustering or Principal Component Analysis (PCA). This has enabled tensor decompositions to extract malicious behavior that has been intentionally obfuscated during our experiments on real network traffic data. In particular, initial experiments have shown that tensor decompositions especially excel at identifying data exfiltration, an activity of special concern in the security community.

We develop and present CANDID, a highly scalable and user-friendly network analysis tool for deeply analyzing network metadata and presenting clear and actionable insights. CANDID is built upon ENSIGN [1], a generic high-performance tensor analysis tool, developed at Reservoir Labs, that provides fast, efficient, and scalable tensor decompositions. ENSIGN provides novel data structures [2] for storing tensors and implementations of multiple tensor decomposition algorithms specifically engineered to scale to large problems [3, 4].

In this paper, we make the following specific contributions:

- Present a scalable network analysis workflow starting from tensor construction from network log data to integrating the results of tensor analysis into widely-used data management platforms such as Splunk.
- Present the use of tensor decompositions in large-scale operational cyber security environments.
- Present concrete, actionable discoveries using this approach.

The remainder of the paper is organized as follows. Section 2 provides an overview of tensors, tensor decompositions, and how to interpret their results. Section 3 discusses some related research work on applying tensor decompositions for cyber security. Section 4 discusses the workflow of our CANDID network analysis tool. Sections 5 and 6 detail the practical deployment of our tool at SC16 SCinet network and Reservoir Labs’ LAN, respectively, and discuss some of the significant results from the deployment. Section 7 discusses our plan for using and demonstrating CANDID/ENSIGN on SC17 SCinet network. Section 8 summarizes our work with a foreword on our ongoing work.

## 2. Tensor Analysis and Tensor Decompositions

### 2.1. Representing Multidimensional Data as Tensors

Tensors (aka multidimensional arrays) are a natural fit for representing data with multiple associated attributes such as network traffic data. Consider a sample data log of network traffic messages. For each message, let us assume that the log records the timestamp of the message, IP address that sent the message, TCP/UDP port that the IP address used, and IP address that the message is sent to. This dataset can be formed into a four-dimensional tensor with the dimensions (“modes” in the tensor analysis parlance) being timestamp, sender IP, receiver IP, and port. For each (timestamp, sender IP, receiver IP, port) tuple, the tensor contains the count of the number of messages sent at that time, by that sender IP, on that port, to the receiver IP.

### 2.2. Tensor Decompositions

Tensor decompositions are a valuable, mathematically sound set of tools for exploratory analysis of multidimensional data and for capturing underlying multidimensional relationships. Tensor decompositions separate the input data into patterns called “components.” Each component represents a latent behavior or correlation from within the dataset. This separation into components occurs without training or upfront specification. There are two prominent tensor decomposition models, namely, CANDECOMP/PARAFAC (CP) decomposition and Tucker decomposition. The particular decomposition model used in this paper is the CP decomposition illustrated in Figure 1.

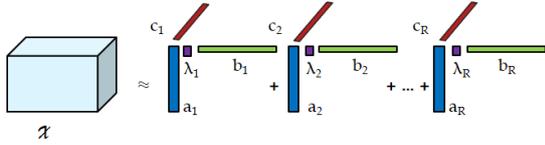


Figure 1: A CP Decomposition of a 3-dimensional tensor into R components.

A CP tensor decomposition decomposes a tensor into a set of components, each of which represents a different pattern extracted from the original tensor. Each component has a weight and then one score vector for each tensor mode. The component weight reflects how large the contribution of this component is to the original dataset. The length of the score vector for the  $i$ th-mode is equal to the number of indices in the  $i$ th-mode.

In the experiments described in this paper, we use the Alternating Poisson Regression (APR) algorithm [5] for computing CP decomposition. This algorithm computes a CP decomposition in a way that is tailored to work on sparse data that is modeled by a Poisson distribution. It is well known that real world count/event data is roughly approximated by a Poisson distribution, and this method produces good decomposition results for cyber data.

### 2.3. Interpreting Decomposition Results

The output components of tensor decompositions are the core of tensor analysis. Let us consider again the previous four-mode tensor with the dimensions timestamp, sender IP, receiver IP, and port. Performing a CP decomposition would decompose this tensor into components. Each component has a weight that corresponds to the volume of network traffic this component describes. This will not be the exact number of messages explained by this component, but an approximate volume (since sometimes components have non-integer weights). Higher weight components correspond to large-scale patterns and low weight components correspond to more anomalous or specific traffic. Each component has four score vectors: one for timestamp, one for sender IP, one for receiver IP, and one for port. The length of the score vector of each mode will be the total number of distinct entities in that mode (i.e., number of distinct time steps, sender IPs, receiver IPs, ports).

The individual score values indicate how much a specific index contributes to this cluster of network activity. Each score is a continuous value between 0.0 and 1.0. Within each score vector, the scores are normalized so that they sum to 1.0. Thus a natural interpretation of the score is what fraction of the total network messages represented in this component this index contributes to.

### 3. Related Work

Some existing work in the literature (e.g., MultiAspectForensics [6] and MalSpot [7]) has applied tensor decompositions to network traffic data in order to extract anomalies and malicious patterns. Unlike the other work, we extend beyond theoretical research and toolbox development to demonstrate practical operational results. Our experiments are set up in operational cyber security environments, enabling us to study how tensor decompositions fit into a realistic work flow. The network at SC16 was extremely high volume and bad actors were common making it a uniquely ideal data source compared to publicly available network datasets.

Since the ENSIGN tensor toolbox includes a scalable implementation of Poisson regression based CP algorithm for tensors of arbitrarily many dimensions, we also avoid many of the limitations existing in other work such as MalSpot. Our Poisson regression based tensor decomposition method produces sparse component vectors with all non-negative scores. This is critical for network security analysis because it allows components to be examined individually for the behavior they represent. With the commonly used CP algorithm (based on Alternating Least Squares method), the component vectors are often dense and have both positive and negative scores. This means that most entries in the reconstructed tensor have contributions from multiple components (some of which might be positive while others may be negative) that complicates analysis. MalSpot, for example, addresses this problem by plotting IP address scores between components and then clustering the points in that space. However, this creates a proliferation of work for the analyst. In the case of highly heterogeneous data such as real network traffic, a day's worth of data might take tens to hundreds of components to accurately decompose. To plot the IP scores between all possible pairs (or even triples) of components is not feasible.

#### 4. CANDID Workflow

Figure 2 represents the CANDID network analysis workflow. CANDID involves a data transformation module that transforms network flow logs (currently Bro [8] logs) into tensors. The data transformation module can build an arbitrary number of tensors from network flow logs.

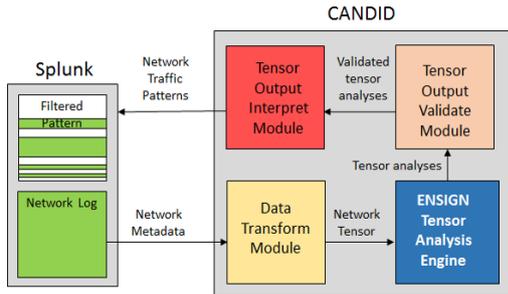


Figure 2: CANDID Workflow

There are multiple Bro log files (such as conn.log, files.log, dns.log, http.log) and each log file has multiple fields or attributes. In theory, one could construct an arbitrarily large number of tensors depending upon the choices made in terms of the log files, log file attributes, and transformations applied on the attributes. We choose an initial suite of network tensors that provides visibility into diverse network traffic patterns. The current version of the data transformation module takes in four different kinds of flow and protocol-specific Bro logs, namely, conn.log, files.log, dns.log, and http.log and produces a suite of tensors (as shown in Table 1) for decomposition that gives maximum network visibility and uncovers network patterns/behaviors and threats.

The tensors created from the network flow logs are analyzed using the ENSIGN tensor analysis engine, the core module of the workflow. The results from tensor analysis are passed on to a tensor output validation module (this module is an optional module in the workflow and is not a focus of this paper), from which the tensor decomposition results are passed on to a Splunk [9] dashboard for a user-friendly visualization of the results. Developing the Splunk dashboard is a work in progress. For the work described in this paper, the Splunk investigation is done manually. However it is to be noted that the components from tensor decompositions (that are few hundreds in number compared

to the million or billion lines of original logs) provide the map to an effective investigation in Splunk that requires less cognitive load.

A network traffic pattern is ultimately a set of network log entries that belong to a distinct class of activity. Therefore we can represent network traffic patterns as filters to apply to the original network logs that select only those messages associated with this pattern. The advantage of doing this within Splunk is that the user/analyst not only has immediate access to all the statistics and visualization tools available, but is also enabled to inspect a single discovered traffic pattern in greater detail. The Splunk dashboard will expose the network patterns to an analyst in a clear and interpretable way that does not require tensor expertise.

#### 5. Tensor Analysis at SC16 SCinet

SCinet, described as “the fastest network connecting the fastest computers,” is set up each year at SC – the International Conference for High Performance Computing, Networking, Storage and Analysis. In 2016 SCinet offered 3.15Tbps from the network operations center and connected over 12,000 researchers. There is no firewall and no asset identification or authorization. We were located in the security operations center (SOC) for SCinet analyzing network traffic metadata from 40 network taps across 1/10/100Gbps wired connections and 200 wireless access points. A cluster of Reservoir Labs’ R-Scope network security appliances [10] running a highly optimized version of the Bro network security monitor provided network traffic metadata. Tensor decompositions were performed on a 12-core HPE Apollo 2000 machine.

Numerous threats and suspicious sessions were isolated into distinct decomposition components. We present a list of interesting network traffic patterns and security attacks that we identified and observed from our tensor analysis. In the subsequent sub-sections, we discuss in detail some of these patterns/behaviors, including security threats/attacks.

We found a number of interesting network behaviors and activities from our connections tensor. Some of them include:

- SSH scanning and SSH password guessing
- Groups of IPs working together to accomplish scans leading to successful SSH infiltration
- Bit-torrenting

Tensor Name	Bro Log	Tensor Dimensions
Connections Tensor	The connections log (conn.log)	Time x Sender IP x Receiver IP x Port
Outgoing Tensor	Connections log entries with local sender and external receiver	Time x Sender IP x Receiver IP x Port
Incoming Tensor	Connections log entries with local receiver and external sender	Time x Sender IP x Receiver IP x Port
Time Independent Connections Tensor	The connections log	Sender IP x Receiver IP x Port x Connection State
Extended Time Independent Connections Tensor	The connections log	Sender IP x Receiver IP x Port x Connection duration x Originator bytes x Connection State
File Transfer Tensor	The file transfer log (files.log)	Time x Sender IP x Receiver IP x MIME-Type
HTTP Tensor	The HTTP traffic log (http.log)	Time x Sender IP x Receiver IP x URI x User Agent
DNS Query Tensor	All queries from the DNS log (dns.log)	Time x Sender IP x Receiver IP x Query x Query Type

Table 1: CANDID Tensor Library

- Internet Printing Protocol traffic indicating a vulnerable machine
- Isolating a timeperiod when a particular machine had a vulnerability through port 51413
- Private network CAPWAP traffic – unusually more common at night than during conference operating hours
- Security team’s Nessus scanner traffic including internal management of the scanner
- Boomerang power monitor traffic
- Steam downloads and scanning of gaming server ports
- Vulnerable Brazilian university supercomputer traffic

Our tensor analysis using extended connections tensor (including metadata attributes such as connection duration and originator bytes), we found and isolated patterns and activities that were obfuscated. Some of them that were security relevant are:

- Anomalous outgoing ICMP traffic indicating exfiltration
- Exfiltration through concealed outgoing DNS traffic
- NTP amplification attack
- Late night outgoing SSH connections to many hosts

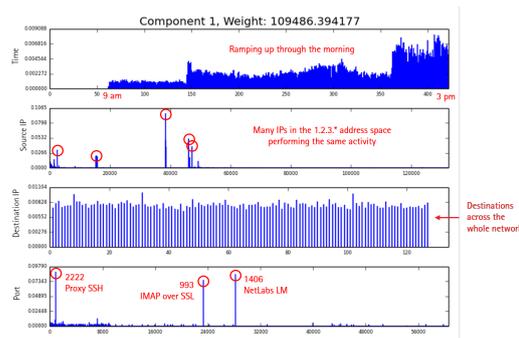


Figure 3: A component from the decomposition of an “incoming traffic” tensor. The component represents distributed network mapping and port scanning with strong likelihood of hostile intent.

The tensor analysis from the DNS tensor identified the following interesting but non-malicious entities:

- DNS hierarchy mapping of a leading company
- Misconfigured DNS server of a popular university
- Outgoing LDAP through DNS “SRV” type requests (not malicious but anomalous)

### 5.1. External Scanners

We now discuss how we captured and isolated the evolution of an external scanning attack on the network and a subsequent data exfiltration from a compromised host.

The component in Figure 3 represents eight hours of traffic from 8AM to 4PM on the opening day of the conference. The topmost chart (timestamp) shows activity starting shortly after the conference start time of 9AM and ramping up throughout the day. The second chart (source IP) shows inbound traffic from a number of external IP addresses (the IP addresses are anonymized as “1.2.3.\*” in the figure). The third chart (destination IP) shows a large number of internal IP addresses as destinations of the inbound traffic. The fourth chart (destination port) shows a specific set of ports being targeted by the external hosts.

It is clear that this component represents a coordinated attempt by multiple external actors to find hosts on SCinet with particular services enabled. In other words, the component shown in Figure 3 represents distributed network mapping and port scanning with strong likelihood of hostile intent. It, indeed, turned out to be the reconnaissance phase of a successful attack.

Further investigation of the IP addresses involved resulted in the discovery of a compromised host targeted during the initial reconnaissance phase. This is captured by the Splunk investigation and tensor decomposition component shown in Figure 4. The tensor decomposition component (Figure 4(b)) shows that the compromised host made 40,000 outgoing SSH connections, which is clearly a very bad sign. The component also shows that the outgoing SSH connections started after the compromise, remained heavy initially, went relatively low-key after sometime, and then resumed heavily after 8 hours.

To summarize, Figures 3 and 4 illustrate the evolution of an attack – an initial component showing distributed network mapping and port scanning and a later component showing promiscuous outgoing SSH traffic from one of the scanned hosts. Splunk was used to connect the dots between the components and confirm the attack. This is one example of a network threat identified using tensor decomposition with no prior attack signature.

### 5.2. ICMP Tunneling

We discuss below a case of suspected ICMP tunneling that was neither seen by other security personnel nor surfaced by other network security tools.

Anomalous ICMP traffic is difficult to distinguish with tensor decompositions done on tensors with standard attributes such as IP addresses, port, connection state/time. Figure 5 shows components

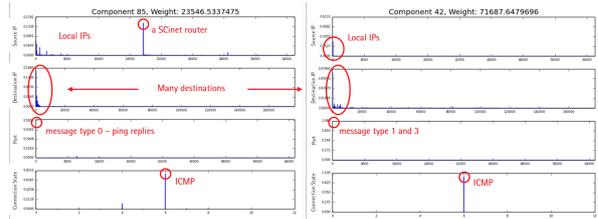


Figure 5: Components from the decomposition of a “basic time independent connections” tensor, showing ICMP traffic patterns from which it is not clear to distinguish anomalous ICMP traffic from normal ICMP traffic.

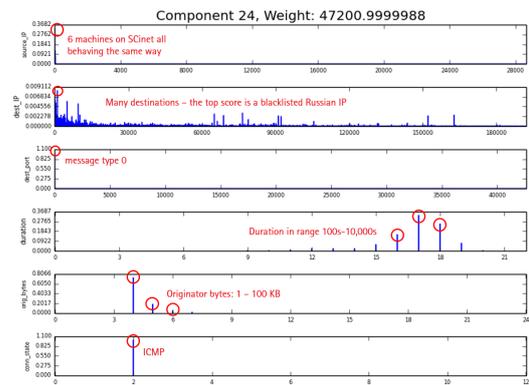


Figure 6: A component from the decomposition of an “extended time independent connections” tensor, clearly identifying suspicious ICMP traffic.

representing traffic patterns involving ICMP messages. As it can be seen, typically ICMP traffic is separated by message type (in Bro logs, the “port” field is overloaded with ICMP message type for ICMP messages). These components represent patterns that look like normal ICMP traffic and do not indicate any abnormal behavior.

We performed subsequent tensor decompositions on tensors constructed from the same data as before, but with additional attributes such as connection duration (binned by log10 scale) and number of originator bytes (binned by log10 scale). The decompositions of these larger tensors with additional metadata attributes immediately (and clearly) uncovered suspicious network traffic involving ICMP. This is shown in Figure 6.

The first chart in Figure 6 (sender IP) revealed six IP addresses and each IP belonged to a different subnet. These IPs did not trigger any signature-based alert, threat intel, or Intrusion Detection System (IDS) alert. The top scoring destination IP (from the second chart) was a blacklisted Russian



Internet. A single R-Scope network security appliance provides network traffic metadata. Tensor decompositions are performed on a 32-core x86 server.

Deep analysis of Reservoir Labs network through CANDID illustrated three main capabilities.

1. Ability to give a high-level overview of the most common traffic on the network.
2. Extraction of patterns relevant to network monitoring and maintenance. Examples include:
  - DHCP misconfiguration
  - Printer misconfiguration
  - DNS caching failure
  - UPnP misconfiguration
  - Noisy HTTPS (change in network state)
  - Link local IPv6 traffic
3. Successful detection of synthetic anomalous traffic seeded into the network.

### 6.1. Network Maintenance and Monitoring

Some of the decomposition patterns from the deep analysis corresponding to background systems management traffic revealed concretely actionable network misconfigurations, highlighting the value of the CANDID approach for network monitoring. Figure 8 enumerates a few such examples.

Figure 8(a) shows a component representing a traffic pattern that occurs regularly through time and entirely on port 67 – corresponding to DHCP traffic. There are two destination IP addresses that have non-zero scores and these correspond to our network’s two DHCP servers. There are a handful of different sender IP addresses including Windows virtual machines (VMs) and several office phones. To further investigate the traffic pattern represented here, we filtered the conn.log in Splunk to only include the timestamps, senders, receivers, and ports with a non-zero score in this tensor component. This revealed DHCP misconfiguration in the Windows VMs and office phone systems identified by the component.

Figure 8(b) shows a component that has one high-scoring sender, two high-scoring receivers, and two high scoring ports. The dominant sender IP address is an office workstation, the two receiver IP addresses are the Epson printer and the broadcast address (255.255.255.255), and finally the two ports are 3289 and 161. The component contains two related activities - Epson printer specific connections

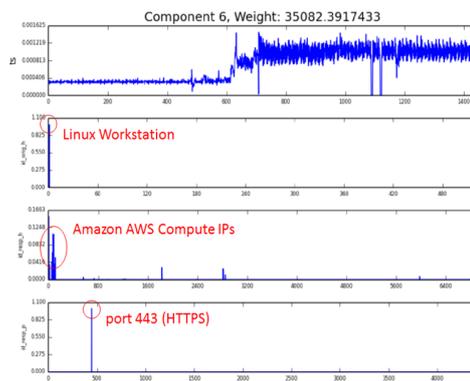


Figure 9: A component showing a sudden increase in traffic – noisy HTTPS traffic

over port 3289 and SNMP connections on port 161. The constant noisy chatter (clearly unrelated to an actual print request) indicates a device misconfiguration. Furthermore, this workstation has not been in active use for several months and is therefore a likely candidate for inspection.

Figure 8(c) represents a component that revealed a DNS caching failure. The component involves a single sender and receiver. The receiver is a DNS server. The sender is a Linux machine that is performing a batch of DNS lookups as part of a research task. This batch of lookups occurs four times within the timeframe of this particular decomposition. However, because of caching misconfiguration, the results are not stored and the enormous batch of lookups has to be repeated each time.

Figure 9 reveals noisy HTTPS traffic. This is not a result of any network misconfiguration. However this demonstrates an important network monitoring capability: detecting a change in network state over time. The top chart (timestamp) in Figure 9 very clearly displays a sudden change around halfway through the day.

The only sender with a non-zero score is a single engineer’s workstation, and the only port with a non-zero score is port 443 (HTTPS/SSL). The top receivers are around 20 different Amazon AWS IP addresses. Upon inspecting the SSL log in Splunk, we found that this activity is almost exclusively lifeyre (an Internet comments service) traffic and the machine’s owner confirmed that this sudden increase in traffic was due to a browser tab left open overnight. This is not malicious activity, but highlights a valuable lesson that the connections tensor with the timestamp dimension can reveal how pat-

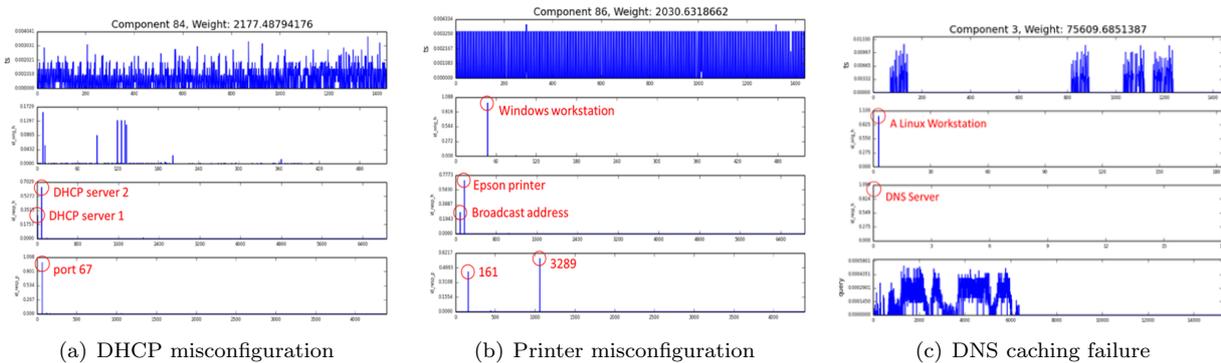


Figure 8: Network misconfigurations detected by CANDID

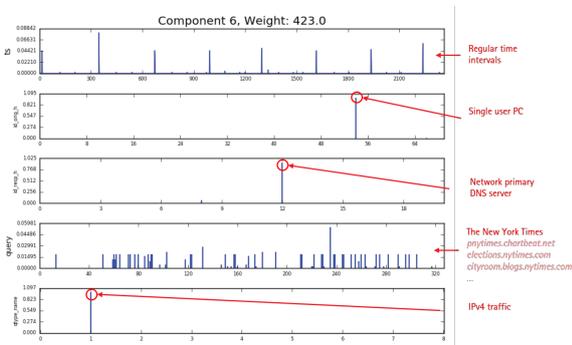


Figure 10: A component with a beaming pattern. In this case a browser left open on the auto-refreshed New York Times home page was identified as the cause.

terms of activity change over time.

### 6.2. Anomalous Behaviors

The analysis revealed many anomalous behaviors that prompted further investigation from the network administrators. We discuss one of the examples in this section. Figure 10 shows a component from our nightly analysis representing an anomalous beaming-like activity. The top chart (timestamp) shows highly regular periodic activity. The second chart (source IP) shows a single user workstation. The third chart (destination IP) shows the primary DNS server. The fourth chart (DNS query) shows a large number of queries for records associated with the New York Times website.

This component prompted the network administrator to investigate it further and it was discovered that a browser left open on the user workstation auto-refreshed New York Times home page and caused the beaming-like traffic.

## 7. Plan for SC17 SCinet

Our research objective at SCinet NRE 2017 is to validate the capability of CANDID/ENSIGN to provide deep network visibility by analyzing diverse network security logs that are available from the SCinet network security stack. As in previous years, we will demonstrate the effectiveness of our network analysis and cyber security capability in an operational cyber environment. We will demonstrate an end-to-end workflow where we take network data from diverse network security logs such as Bro logs, Attivo Networks logs, and other available system logs from the security stack, perform advanced network analysis, and display the identified network patterns, indicating attacks or interesting network behaviors, in a Splunk dashboard.

At SCinet NRE 2015 and 2016, we successfully demonstrated our tool on offline network data feeds. Specifically, we demonstrated how our tool separated normal and off-normal traffic patterns in a way that led to the discovery of indicators consistent with, and in some cases prior to, human analyst discovery (e.g., a distributed takeover attack on a vendor booth and suspected ICMP-based data exfiltration). The major advance that we plan to demonstrate at SCinet NRE 2017 is the addition of a streaming analysis capability for cyber security that will improve the timeliness of the previously demonstrated offline analysis of network metadata. Further, we plan to demonstrate how information from multiple network security sources could be fused to provide a comprehensive and coordinated view of the network behaviors.

We will install and operate CANDID/ENSIGN from a node in the SCinet Network Security Cloud that is provided by CloudLab. We will use Splunk

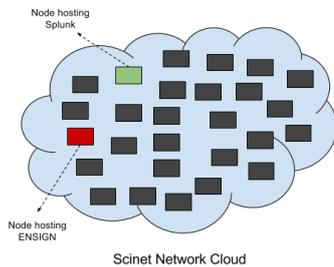


Figure 11: Pictorial representation of SCinet cloud infrastructure and deployment of CANDID/ENSIGN in the cloud.

as a data management and visualization platform. We will pull data from the Splunk instance hosted in the Cloud and use CANDID Splunk dashboard to view and show results. Figure 11 shows the network topology.

## 8. Conclusion

CANDID, along with the foundational work of ENSIGN, contributes uniquely to network traffic analysis by solving many of the problems not addressed by traditional approaches for cyber security. Avoiding requirements for large training datasets that plague many machine learning techniques and not requiring a definition of normal or abnormal activity upfront are some of the clear benefits that make this approach elegant and powerful. Furthermore, CANDID/ENSIGN tensor decompositions capture patterns that span the entire multidimensional space extracting anomalous activity even when an attacker explicitly crafts their actions to subvert signature-based detection methods, or hides attacks within a broader pattern of normal activity. The practical use of the tool in real operational cyber security environments on large-scale networks is proof that it can be integrated into a network security specific solution.

Our plan for ongoing and future work is to further improve and advance the tool with respect to cyber security practice and move closer to a deployable, turnkey solution. Some of the work towards this objective include automatic classification of network

traffic patterns and threats, automatic distinction of malicious and benign anomalies, and ranking of threats and/or interesting patterns.

## Acknowledgment

This material is partially based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award Number DE-SC0017081.

## References

- [1] R. Labs, ENSIGN Tensor Toolbox.  
URL <https://www.reservoir.com/product/ensign-cyber>
- [2] M. Baskaran, B. Meister, N. Vasilache, R. Lethin, Efficient and Scalable Computations with Sparse Tensors, in: IEEE High Performance Extreme Computing Conference, Waltham, MA, 2012.
- [3] M. Baskaran, B. Meister, R. Lethin, Low-overhead Load-balanced Scheduling for Sparse Tensor Computations, in: IEEE High Performance Extreme Computing Conference, Waltham, MA, 2014.
- [4] M. Baskaran, T. Henretty, B. Pradelle, M. H. Langston, D. Bruns-Smith, J. Ezick, R. Lethin, Memory-efficient Parallel Tensor Decompositions, in: IEEE High Performance Extreme Computing Conference, Waltham, MA, 2017.
- [5] E. C. Chi, T. G. Kolda, On Tensors, Sparsity, and Non-negative Factorizations, arXiv:1304.4964 [math.NA] (December 2011).  
URL <http://arxiv.org/abs/1112.2414>
- [6] K. Maruhashi, F. Guo, C. Faloutsos, Multiaspectforensics: Pattern mining on large-scale heterogeneous networks with tensor analysis, in: International Conference on Advances in Social Networks Analysis and Mining, Kaohsiung, Taiwan, 2011.
- [7] H.-H. Mao, W. Chung-Jung, E. E. Papalexakis, K.-C. L. Christos Faloutsos, T.-C. Kao, Malspot: Multi2 malicious network behavior patterns analysis, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Tainan, Taiwan, 2014.
- [8] V. Paxson, Bro: A system for detecting network intruders in real-time, *Comput. Netw.* 31 (23-24) (1999) 2435–2463. doi:10.1016/S1389-1286(99)00112-7. URL [http://dx.doi.org/10.1016/S1389-1286\(99\)00112-7](http://dx.doi.org/10.1016/S1389-1286(99)00112-7)
- [9] Splunk, Server and Network Infrastructure Monitoring. URL [https://www.splunk.com/en\\_us/solutions/solution-areas/it-operations-management/server-and-network-infrastructure-monitoring.html](https://www.splunk.com/en_us/solutions/solution-areas/it-operations-management/server-and-network-infrastructure-monitoring.html)
- [10] R. Labs, R-Scope.  
URL <https://www.reservoir.com/product/r-scope>